



Available at

www.ElsevierComputerScience.com

POWERED BY SCIENCE @ DIRECT®

Theoretical Computer Science 309 (2003) 357–380

Theoretical
Computer Sciencewww.elsevier.com/locate/tcs

Structured theories and institutions[☆]

Francisco Durán^a, José Meseguer^{b,*}^a*Dpto. de Lenguajes y Ciencias de la Computación, Universidad de Málaga, Málaga 29079, Spain*^b*Computer Science Department, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*

Received 30 May 2002; accepted 22 April 2003

Communicated by M. Wirsing

Abstract

Category theory provides an excellent foundation for studying structured specifications and their composition. For example, theories can be structured together in a diagram, and their composition can be obtained as a colimit. There is, however, a growing awareness, both in theory and in practice, that structured theories should not be viewed just as the “scaffolding” used to build unstructured theories: they should become *first-class citizens* in the specification process. Given a logic formalized as an institution \mathcal{I} , we therefore ask whether there is a good definition of the category of structured \mathcal{I} -theories, and whether they can be naturally regarded as the ordinary theories of an appropriate institution $\mathcal{S}(\mathcal{I})$ generalizing the original institution \mathcal{I} . We answer both questions in the affirmative, and study good properties of the institution \mathcal{I} inherited by $\mathcal{S}(\mathcal{I})$. We show that, under natural conditions, a number of important properties are indeed inherited, including cocompleteness of the category of theories, liberality, and extension of the basic framework by freeness constraints. The results presented here have been used as a foundation for the module algebra of the Maude language, and seem promising as a semantic basis for a generic module algebra that could be both specified and executed within the logical framework of rewriting logic.

© 2003 Elsevier B.V. All rights reserved.

Keywords: Structured theories; Institutions; Maude

1. Introduction

Structuring mechanisms are vital means for reusing software and for mastering the complexity of large systems at all levels, including specifications and code. Category

[☆] Research supported by DARPA through AIR Force Research Laboratory Contract F30602-02-C-0130, ONR Grant N00014-02-1-0715 and CICYT projects TIC 2000-0701-C02-01 and TIC 2001-2705-C03-02.

* Corresponding author.

E-mail addresses: duzan@lcc.uma.es (F. Durán), meseguer@cs.uiuc.edu (J. Meseguer).

theory provides an excellent foundation for studying structured specifications and their composition. A key contribution in the late 1970s and early 1980s was made by Burstall and Goguen with the Clear [4] specification language, that proposed taking colimits of theories as a systematic way of “putting theories together.” Clear was based on many-sorted equational logic, but its categorical semantics was in fact logic-independent. This led Goguen and Burstall to propose the notion of *institution* as an axiomatization of a general logic, and to generalize the Clear-like operations to institutions [18,19]. These ideas have had a great theoretical and practical impact: see Refs. [3,17], the survey [22], and the literature on logic-independent specification building operations, e.g. [2,10,27,29].

Typically, theory composition operations begin with theories structured in some way—for example, a diagram—and result in an unstructured, or less structured, specification as their result—for example, a colimit. That is, structured theories are often “flattened” when being composed. There are however good reasons for preserving their structure. Besides the obvious understandability and design documentation reasons, it is often very useful to consider theory-building operations whose results are structured theories. For example, refining a software design can be best understood as refining structured theories [32]; also, even when we may want to extract a flattened theory, it can be much more efficient to operate at the level of structured theories [12,13]. There are also more intrinsic reasons, namely, when the semantics associated to a structured module *essentially depends on its structure*. For example, we often want to associate to the inclusion of a parameter theory into the body of a parameterized specification a *freeness constraint*, requiring that the models of the body are free extensions of the models of the parameter; more generally, one can similarly consider other notions of constraint [4,16,19,28,30]. In practice, the need for keeping and using structure is both recognized and supported by a number of languages and systems such as, for example, languages in the Clear/OBJ tradition [4,6,9,20], SPECWARE [32], and CASL [8].

Although a number of concepts and techniques have been suggested both at the theoretical and specification language levels to keep and use the necessary amount of structure for specific purposes, the most satisfactory way of addressing the need for preserving structure is to make structured theories *first-class citizens*. In the categorical spirit, this leads to seeking a good definition of the category of structured theories, and to investigating whether structured theories can naturally be regarded as the ordinary theories of an appropriate institution. The most basic form of structured theory is that of a *hierarchy* of theory inclusions, in the sense that more complex forms of structured theories can often be *normalized* to hierarchies [12,13], perhaps keeping some additional information such as freeness constraints. Hierarchies are of course special kinds of *diagrams*, and this suggests using categories of diagrams and categorical constructions on diagrams as the theoretical basis.

The use of diagrams for structuring purposes has also been emphasized by other authors. In a limited form they were used in Clear to deal with shared structure in categorical constructions by means of *based theories* [4]. Diagrams are first-class citizens in SPECWARE [32], and are used to structure and refine specifications; furthermore, an appropriate diagram category is defined in such a way that a colimit-like functor yields an operation of horizontal composition satisfying, by functoriality, the

expected laws of compatibility between horizontal and vertical composition [32]. Based on the SPECWARE ideas, Dimitrakos has proposed a way of parameterizing specifications by diagrams of specifications, and of inducing an instantiation by means of a family of parallel instantiating morphisms whose sources are the components of the parameter diagram [11].

In this paper, which is an extended version of [14], we address a number of issues about structured theories that, as far as we know, have not been systematically studied before. The most basic issue is: given an institution \mathcal{I} , can we naturally associate to it another institution $\mathcal{S}(\mathcal{I})$ whose ordinary theories are the structured theories of \mathcal{I} ? We answer this question in the affirmative, and then proceed to study to what extent good properties of the institution \mathcal{I} are also inherited by $\mathcal{S}(\mathcal{I})$. We show that, under natural conditions, a number of important properties are indeed inherited, including cocompleteness of the category of theories, liberality, and extension of the basic framework by freeness constraints.

We have used the present work as the theoretical foundation for the module algebra of the Maude specification language [6]. In this module algebra, structured theories are first-class citizens, and module operations result in other structured theories [12,13]. Using the fact that rewriting logic is reflective [5,7], the entire module algebra is both specified and executed within the logic of Maude [12]. As we further explain in the conclusions, using the logic-independent semantics for structured theory compositions developed in this paper and the logical framework properties of Maude [23], we plan to generalize Maude's module algebra to an executable *generic module algebra* that could be instantiated for any suitable logic represented in the framework.

The rest of the paper is organized as follows. Section 2 reviews some basic definitions about institutions; Section 3 gives basic results about categories of diagrams; Section 4 presents our main definitions and results about the institution $\mathcal{S}(\mathcal{I})$ of structured \mathcal{I} -theories and its properties; Section 5 illustrates the use of the categorical constructions with several examples; and Section 6 offers some concluding remarks.

2. Institutions

The theory of institutions [19] allows us to discuss the relationship between theories and models without committing ourselves to a particular logical inference system.

Definition 1 (Goguen and Burstall [18]). An *institution* \mathcal{I} is a 4-tuple $(\mathbf{Sign}_{\mathcal{I}}, \text{sen}_{\mathcal{I}}, \text{Mod}_{\mathcal{I}}, \models)$ such that:

- $\mathbf{Sign}_{\mathcal{I}}$ is a category whose objects are called *signatures*,
- $\text{sen}_{\mathcal{I}} : \mathbf{Sign}_{\mathcal{I}} \rightarrow \mathbf{Set}$ is a functor associating to each signature Σ a set of Σ -sentences,
- $\text{Mod}_{\mathcal{I}} : \mathbf{Sign}_{\mathcal{I}} \rightarrow \mathbf{Cat}^{\text{op}}$ is a functor mapping each signature Σ to a category whose objects are called Σ -models, and
- \models is a function associating to each $\Sigma \in |\mathbf{Sign}_{\mathcal{I}}|$ a binary relation $\models_{\Sigma} \subseteq |\text{Mod}_{\mathcal{I}}(\Sigma)| \times \text{sen}_{\mathcal{I}}(\Sigma)$ called *satisfaction*, in such a way that the following property holds for any $M' \in |\text{Mod}_{\mathcal{I}}(\Sigma')|$, $H : \Sigma \rightarrow \Sigma'$, $\varphi \in \text{sen}_{\mathcal{I}}(\Sigma)$:

$$M' \models_{\Sigma'} \text{sen}_{\mathcal{I}}(H)(\varphi) \iff \text{Mod}_{\mathcal{I}}(H)(M') \models_{\Sigma} \varphi.$$

Given a signature Σ , a presentation of a theory is given by a set Γ of Σ -sentences. We can therefore denote a theory presentation as a pair (Σ, Γ) . Given a presentation (Σ, Γ) , we define the category $\text{Mod}_{\mathcal{J}}(\Sigma, \Gamma)$ as the full subcategory of $\text{Mod}_{\mathcal{J}}(\Sigma)$ determined by those models $M \in |\text{Mod}_{\mathcal{J}}(\Sigma)|$ that satisfy all the sentences in Γ , i.e., $M \models_{\Sigma} \varphi$ for all $\varphi \in \Gamma$.

We can extend the satisfaction relation to sets of sentences as follows:

$$M \models_{\Sigma} \Gamma \text{ iff } M \models_{\Sigma} \varphi \text{ for all } \varphi \in \Gamma.$$

Then, the relation between sets of sentences and sentences given by

$$\Gamma \models_{\Sigma} \varphi \text{ iff } M \models_{\Sigma} \varphi \text{ for each } M \in |\text{Mod}_{\mathcal{J}}(\Sigma, \Gamma)|$$

allows us to associate to an institution an entailment system in the sense of [24]. For any signature Σ , the *closure* of a set Γ of Σ -sentences is $\Gamma^{\bullet} = \{\varphi \mid \Gamma \models_{\Sigma} \varphi\}$. The Σ -theory presented by (Σ, Γ) is then given by $(\Sigma, \Gamma^{\bullet})$.

Given presentations of theories (Σ, Γ) and (Σ', Γ') , a theory morphism $H : (\Sigma, \Gamma) \rightarrow (\Sigma', \Gamma')$ is a signature morphism $H : \Sigma \rightarrow \Sigma'$ such that if $\varphi \in \Gamma$ then $\text{sen}_{\mathcal{J}}(H)(\varphi) \in \Gamma'^{\bullet}$, that is, for all $\varphi \in \Gamma$, $\Gamma' \models_{\Sigma'} \text{sen}_{\mathcal{J}}(H)(\varphi)$.

Definition 2. Given an institution \mathcal{J} , its category $\mathbf{Th}_{\mathcal{J}}$ of *theories* has as objects presentations of theories (Σ, Γ) and as arrows theory morphisms.¹ We denote by $\text{sign}_{\mathcal{J}} : \mathbf{Th}_{\mathcal{J}} \rightarrow \mathbf{Sign}_{\mathcal{J}}$ the forgetful functor sending each theory to its underlying signature.

For any institution \mathcal{J} , the model functor $\text{Mod}_{\mathcal{J}} : \mathbf{Sign}_{\mathcal{J}} \rightarrow \mathbf{Cat}^{\text{op}}$ extends to a functor $\text{Mod}_{\mathcal{J}} : \mathbf{Th}_{\mathcal{J}} \rightarrow \mathbf{Cat}^{\text{op}}$, by mapping a theory (Σ, Γ) to the full subcategory $\text{Mod}_{\mathcal{J}}(\Sigma, \Gamma)$ of $\text{Mod}_{\mathcal{J}}(\Sigma)$. The institution \mathcal{J} is called *liberal* if for each theory morphism $H : (\Sigma, \Gamma) \rightarrow (\Sigma', \Gamma')$ the functor $\text{Mod}_{\mathcal{J}}(H)$ has a left adjoint. We call \mathcal{J} *exact* if $\text{Mod}_{\mathcal{J}} : \mathbf{Th}_{\mathcal{J}} \rightarrow \mathbf{Cat}^{\text{op}}$ preserves colimits.

3. Diagram categories

The issue of whether the category $Dg(\mathcal{C})$ of diagrams over a category \mathcal{C} has colimits is important, because for $\mathcal{C} = \mathbf{Sign}_{\mathcal{J}}$ this specializes to colimits of structured signatures, which can then be used to define colimits of structured theories. We show in this section that, if \mathcal{C} is cocomplete, then $Dg(\mathcal{C})$ is also cocomplete. This is probably a “folklore” result. Since we are not aware of a suitable textbook exposition to give as a reference, we include the details here to make the paper self-contained.

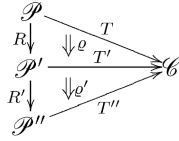
Given a cocomplete category \mathcal{C} and a small category \mathcal{A} , the category of functors from \mathcal{A} to \mathcal{C} , which we denote by $\mathcal{C}^{\mathcal{A}}$, is also a cocomplete category [21]. Furthermore, in $\mathcal{C}^{\mathcal{A}}$ all colimits can be constructed pointwise.

¹ Note that the objects of $\mathbf{Th}_{\mathcal{J}}$ are *presentations* of theories. We follow here the terminology of general logics [24], instead of Goguen and Burstall’s original definition [19]. In what follows, when we talk about a theory (Σ, Γ) we shall mean a theory presentation.

Theorem 3 (Left Kan extensions [21,31]). Let \mathcal{B} be a small category, let $F: \mathcal{B} \rightarrow \mathcal{D}$ be a functor, and let \mathcal{C} be a cocomplete category. Then, the functor $\tilde{F} = \mathcal{C}^F: \mathcal{C}^{\mathcal{B}} \rightarrow \mathcal{C}^{\mathcal{D}}$ has a left adjoint $\tilde{V}_F: \mathcal{C}^{\mathcal{D}} \rightarrow \mathcal{C}^{\mathcal{B}}$, called the left Kan extension along F .

Definition 4 (Schubert [31]). Let \mathcal{C} be a category. The *diagram category* $Dg(\mathcal{C})$ has as objects functors $T: \mathcal{P} \rightarrow \mathcal{C}$, where \mathcal{P} is a small category. If $T: \mathcal{P} \rightarrow \mathcal{C}$ and $T': \mathcal{P}' \rightarrow \mathcal{C}$ are objects, then a morphism $(R, \varrho): T \rightarrow T'$ consists of a functor $R: \mathcal{P} \rightarrow \mathcal{P}'$ and a natural transformation $\varrho: T \rightarrow T' \cdot R$.

The composition of morphisms (R, ϱ) and (R', ϱ') , as depicted in the diagram below, is given by the morphism $(R' \cdot R, \varrho' R \cdot \varrho)$.



Theorem 5. Let \mathcal{C} be a cocomplete category. Then, the category $Dg(\mathcal{C})$ is also cocomplete.

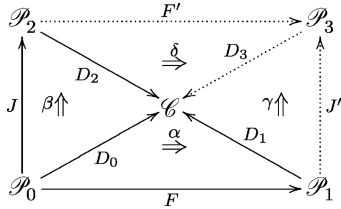
Since a category with pushouts and coproducts has all colimits, we split the proof of Theorem 5 in two separate lemmas.²

Lemma 6. If \mathcal{C} is cocomplete, then $Dg(\mathcal{C})$ has pushouts.

Proof. Given small categories $\mathcal{P}_0, \mathcal{P}_1$, and \mathcal{P}_2 , diagrams D_0 in $\mathcal{C}^{\mathcal{P}_0}$, D_1 in $\mathcal{C}^{\mathcal{P}_1}$, and D_2 in $\mathcal{C}^{\mathcal{P}_2}$, and diagram morphisms $(F, \alpha): D_0 \rightarrow D_1$ and $(J, \beta): D_0 \rightarrow D_2$, we need to construct a pushout object D_3 in $\mathcal{C}^{\mathcal{P}_3}$ (for the appropriate \mathcal{P}_3) and corresponding morphisms in $Dg(\mathcal{C})$, as depicted in the following diagram:

$$\begin{array}{ccc} D_2 & \xrightarrow{(F', \delta)} & D_3 \\ (J, \beta) \uparrow & & \uparrow (J', \gamma) \\ D_0 & \xrightarrow{(F, \alpha)} & D_1 \end{array}$$

In more detail, the above diagram can be depicted as follows.



² We prefer to use pushouts instead of coequalizers because of their extensive applications to parameterized theories.

First, we define the small category \mathcal{P}_3 , with $J': \mathcal{P}_1 \rightarrow \mathcal{P}_3$ and $F': \mathcal{P}_2 \rightarrow \mathcal{P}_3$, as the pushout of F and J in **Cat**. The intuitive idea in order to build up the desired diagram D_3 is the following: if the D_i were all in the same category $\mathcal{C}^{\mathcal{P}}$, then the pushout could be constructed pointwise. We obtain the more general construction by taking the Kan extensions along the corresponding functors to \mathcal{P}_3 for each of these diagrams, thus “moving” them all to $\mathcal{C}^{\mathcal{P}_3}$; D_3 is then the pushout of the Kan-extended diagrams in $\mathcal{C}^{\mathcal{P}_3}$, which can be computed pointwise.

The morphism $F: \mathcal{P}_0 \rightarrow \mathcal{P}_1$ gives us the forgetful functor $\tilde{F}: \mathcal{C}^{\mathcal{P}_1} \rightarrow \mathcal{C}^{\mathcal{P}_0}$ by simply composing with F , that is, $\tilde{F}D_1 = D_1 \cdot F$. By Theorem 3, there exists a left adjoint to \tilde{F} , denoted \tilde{V}_F . The adjunction is then given as

$$\langle \tilde{V}_F, \tilde{F}, \eta^F, \varepsilon^F \rangle: \mathcal{C}^{\mathcal{P}_0} \rightleftarrows \mathcal{C}^{\mathcal{P}_1},$$

where η^F and ε^F are, respectively, the unit and the counit. Given, for example, a diagram D_0 in $\mathcal{C}^{\mathcal{P}_0}$, we denote by $\eta_{D_0}^F$ the component of η^F corresponding to D_0 . By definition of \tilde{F} , the natural transformation $\alpha: D_0 \rightarrow D_1 \cdot F$ in $\mathcal{C}^{\mathcal{P}_0}$ can be rewritten as $\alpha: D_0 \rightarrow \tilde{F}D_1$, and, because of the adjunction $\tilde{V}_F \dashv \tilde{F}$, there is a unique arrow $\tilde{\alpha}: \tilde{V}_F D_0 \rightarrow D_1$ in $\mathcal{C}^{\mathcal{P}_1}$ such that $\tilde{F}\tilde{\alpha} \cdot \eta_{D_0}^F = \alpha$.

The same reasoning with the morphism (J, β) gives the adjunction

$$\langle \tilde{V}_J, \tilde{J}, \eta^J, \varepsilon^J \rangle: \mathcal{C}^{\mathcal{P}_0} \rightleftarrows \mathcal{C}^{\mathcal{P}_2},$$

with \tilde{V}_J the left adjoint to the forgetful functor \tilde{J} , having unit and counit η^J and ε^J , respectively. A natural transformation $\beta: D_0 \rightarrow \tilde{J}D_2$ in $\mathcal{C}^{\mathcal{P}_0}$ induces a unique $\tilde{\beta}: \tilde{V}_J D_0 \rightarrow D_2$ in $\mathcal{C}^{\mathcal{P}_2}$ such that $\tilde{J}\tilde{\beta} \cdot \eta_{D_0}^J = \beta$.

Similarly, the functors F' and J' define forgetful functors \tilde{F}' and \tilde{J}' , with $\tilde{F} \cdot \tilde{J}' = \tilde{J} \cdot \tilde{F}'$, because $F' \cdot J = J' \cdot F$. Among all the isomorphic adjoints to each one of them, we can choose $\tilde{V}_{F'}$ and $\tilde{V}_{J'}$ such that $\tilde{V}_{F'} \cdot \tilde{V}_J = \tilde{V}_{J'} \cdot \tilde{V}_F$. So, the following two square diagrams, one of forgetful functors and another of their left adjoints, commute:

$$\begin{array}{ccc} \mathcal{C}^{\mathcal{P}_2} & \xrightleftharpoons[\tilde{F}']{\tilde{V}_{F'}} & \mathcal{C}^{\mathcal{P}_3} \\ \tilde{V}_J \uparrow \tilde{J} & & \tilde{J}' \uparrow \tilde{V}_{J'} \\ \mathcal{C}^{\mathcal{P}_0} & \xrightleftharpoons[\tilde{V}_F]{\tilde{F}} & \mathcal{C}^{\mathcal{P}_1} \end{array}$$

For the new adjunctions $\tilde{V}_{F'} \dashv \tilde{F}'$ and $\tilde{V}_{J'} \dashv \tilde{J}'$, we can choose their units in such a way that the units of the two composite adjunctions³ $\tilde{V}_{F'} \cdot \tilde{V}_J \dashv \tilde{J} \cdot \tilde{F}'$ and $\tilde{V}_{J'} \cdot \tilde{V}_F \dashv \tilde{F} \cdot \tilde{J}'$ are identical. Therefore, given the new adjunctions

$$\langle \tilde{V}_{F'}, \tilde{F}', \eta^{F'}, \varepsilon^{F'} \rangle: \mathcal{C}^{\mathcal{P}_2} \rightleftarrows \mathcal{C}^{\mathcal{P}_3}$$

³ [21, Section IV.8, Theorem 1] Given two adjunctions $\langle F, G, \eta, \varepsilon \rangle: X \rightleftarrows A$ and $\langle \tilde{F}, \tilde{G}, \tilde{\eta}, \tilde{\varepsilon} \rangle: A \rightleftarrows D$, the composite functors yield an adjunction

$$\langle \tilde{F} \cdot F, G \cdot \tilde{G}, G\tilde{\eta}_F \cdot \eta, \tilde{\varepsilon} \cdot \tilde{F}\eta_{\tilde{G}} \rangle: X \rightleftarrows D.$$

and

$$\langle \tilde{V}_{J'}, \tilde{J}', \eta^{J'}, \varepsilon^{J'} \rangle: \mathcal{C}^{\mathcal{P}_1} \rightarrow \mathcal{C}^{\mathcal{P}_3},$$

we have composite adjunctions

$$\langle \tilde{V}_{J'} \cdot \tilde{V}_F, \tilde{F} \cdot \tilde{J}', \tilde{F}\eta_{\tilde{V}_F}^{J'} \cdot \eta^F, \varepsilon^{J'} \cdot \tilde{V}_{J'}\varepsilon_{J'}^F \rangle: \mathcal{C}^{\mathcal{P}_0} \rightarrow \mathcal{C}^{\mathcal{P}_3}$$

and

$$\langle \tilde{V}_{F'} \cdot \tilde{V}_J, \tilde{J} \cdot \tilde{F}', \tilde{J}\eta_{\tilde{V}_J}^{F'} \cdot \eta^J, \varepsilon^{F'} \cdot \tilde{V}_{F'}\varepsilon_{F'}^J \rangle: \mathcal{C}^{\mathcal{P}_0} \rightarrow \mathcal{C}^{\mathcal{P}_3},$$

with $\tilde{F} \cdot \tilde{J}' = \tilde{J} \cdot \tilde{F}'$, $\tilde{V}_{J'} \cdot \tilde{V}_F = \tilde{V}_{F'} \cdot \tilde{V}_J$, $\tilde{F}\eta_{\tilde{V}_F}^{J'} \cdot \eta^F = \tilde{J}\eta_{\tilde{V}_J}^{F'} \cdot \eta^J$, and $\varepsilon^{J'} \cdot \tilde{V}_{J'}\varepsilon_{J'}^F = \varepsilon^{F'} \cdot \tilde{V}_{F'}\varepsilon_{F'}^J$.

Applying $\tilde{V}_{J'}$ to $\tilde{\alpha}$ and $\tilde{V}_{F'}$ to $\tilde{\beta}$, we can move $\tilde{\alpha}$ and $\tilde{\beta}$ to $\mathcal{C}^{\mathcal{P}_3}$, obtaining the morphisms $\tilde{V}_{J'}\tilde{\alpha}: \tilde{V}_{J'}\tilde{V}_F D_0 \rightarrow \tilde{V}_{J'}D_1$ and $\tilde{V}_{F'}\tilde{\beta}: \tilde{V}_{F'}\tilde{V}_J D_0 \rightarrow \tilde{V}_{F'}D_2$. Since $\tilde{V}_{F'}\tilde{V}_J D_0 = \tilde{V}_{J'}\tilde{V}_F D_0$, we can take the pushout of $\tilde{V}_{J'}\tilde{\alpha}$ and $\tilde{V}_{F'}\tilde{\beta}$ in $\mathcal{C}^{\mathcal{P}_3}$, which, since \mathcal{C} is co-complete, can be constructed pointwise [21].

$$\begin{array}{ccc} \tilde{V}_{F'}D_2 & \xrightarrow{\tilde{\delta}} & D_3 \\ \tilde{V}_{F'}\tilde{\beta} \uparrow & p.o. & \uparrow \tilde{\gamma} \\ \tilde{V}_{F'}\tilde{V}_J D_0 & = & \tilde{V}_{J'}\tilde{V}_F D_0 \xrightarrow{\tilde{V}_{J'}\tilde{\alpha}} \tilde{V}_{J'}D_1 \end{array}$$

Given these $\tilde{\delta}: \tilde{V}_{F'}D_2 \rightarrow D_3$ and $\tilde{\gamma}: \tilde{V}_{J'}D_1 \rightarrow D_3$ in $\mathcal{C}^{\mathcal{P}_3}$, there are unique arrows $\delta: D_2 \rightarrow \tilde{F}'D_3$ in $\mathcal{C}^{\mathcal{P}_2}$, and $\gamma: D_1 \rightarrow \tilde{J}'D_3$ in $\mathcal{C}^{\mathcal{P}_1}$, such that $\varepsilon_{D_3}^{F'} \cdot \tilde{V}_{F'}\delta = \tilde{\delta}$ and $\varepsilon_{D_3}^{J'} \cdot \tilde{V}_{J'}\gamma = \tilde{\gamma}$. We claim that this object D_3 in $\mathcal{C}^{\mathcal{P}_3}$ together with the morphisms (F', δ) and (J', γ) constitutes a pushout of (F, α) and (J, β) . To prove this claim, we must show first that the square diagram commutes, and then that it is universal. We sketch here these proofs; the remaining details can be found in [12].

Commutativity. Given the adjunction $\tilde{V}_{J'} \dashv \tilde{F} \cdot \tilde{J}'$ with counit $\varepsilon^{J'} \cdot \tilde{V}_{J'}\varepsilon_{J'}^F$, and given the morphism $\tilde{\gamma} \cdot \tilde{V}_{J'}\tilde{\alpha}: \tilde{V}_{J'}\tilde{V}_F D_0 \rightarrow D_3$ in $\mathcal{C}^{\mathcal{P}_3}$, there is a unique morphism $\tilde{F}\gamma \cdot \alpha: D_0 \rightarrow \tilde{F}\tilde{J}'D_3$ in $\mathcal{C}^{\mathcal{P}_0}$ with $\varepsilon_{D_3}^{J'} \cdot \tilde{V}_{J'}\varepsilon_{\tilde{J}'D_3}^F \cdot \tilde{V}_{J'}\tilde{V}_F(\tilde{F}\gamma \cdot \alpha) = \tilde{\gamma} \cdot \tilde{V}_{J'}\tilde{\alpha}$. In the same way, given the adjunction $\tilde{V}_{F'} \dashv \tilde{J} \cdot \tilde{F}'$ with counit $\varepsilon^{F'} \cdot \tilde{V}_{F'}\varepsilon_{F'}^J$, and given the morphism $\tilde{\delta} \cdot \tilde{V}_{F'}\tilde{\beta}: \tilde{V}_{F'}\tilde{V}_J D_0 \rightarrow D_3$ in $\mathcal{C}^{\mathcal{P}_3}$, there is a unique morphism $\tilde{J}\delta \cdot \beta: D_0 \rightarrow \tilde{J}\tilde{F}'D_3$ in $\mathcal{C}^{\mathcal{P}_0}$ with $\varepsilon_{D_3}^{F'} \cdot \tilde{V}_{F'}\varepsilon_{\tilde{F}'D_3}^J \cdot \tilde{V}_{F'}\tilde{V}_J(\tilde{J}\delta \cdot \beta) = \tilde{\delta} \cdot \tilde{V}_{F'}\tilde{\beta}$.

Since we have chosen the left adjoints such that $\tilde{V}_{J'} \cdot \tilde{V}_F = \tilde{V}_{F'} \cdot \tilde{V}_J$ and the counits such that $\varepsilon^{F'} \cdot \tilde{V}_{F'}\varepsilon_{F'}^J = \varepsilon^{J'} \cdot \tilde{V}_{J'}\varepsilon_{J'}^F$, then, because $\tilde{\delta} \cdot \tilde{V}_{F'}\tilde{\beta} = \tilde{\gamma} \cdot \tilde{V}_{J'}\tilde{\alpha}$, these unique morphisms given by the adjunction must coincide. Therefore, $\tilde{J}\delta \cdot \beta = \tilde{F}\gamma \cdot \alpha$. Together with the fact that $F' \cdot J = F \cdot J'$ by the pushout construction, this means that $(F', \delta) \cdot (J, \beta) = (J', \gamma) \cdot (F, \alpha)$.

Universal property. First, since $F' : \mathcal{P}_2 \rightarrow \mathcal{P}_3$ and $J' : \mathcal{P}_1 \rightarrow \mathcal{P}_3$ are a pushout of $F : \mathcal{P}_0 \rightarrow \mathcal{P}_1$ and $J : \mathcal{P}_0 \rightarrow \mathcal{P}_2$ in **Cat**, there exists a unique functor $K : \mathcal{P}_3 \rightarrow \mathcal{P}_4$ such that $K \cdot J' = K_1$ and $K \cdot F' = K_2$. The functors K , K_1 , and K_2 generate forgetful functors \tilde{K} , \tilde{K}_1 , and \tilde{K}_2 with $\tilde{J}' \cdot \tilde{K} = \tilde{K}_1$ and $\tilde{F}' \cdot \tilde{K} = \tilde{K}_2$. We can choose their left adjoints (Kan extensions), \tilde{V}_K , \tilde{V}_{K_1} , and \tilde{V}_{K_2} , which exist because of Theorem 3, in such a way that $\tilde{V}_J \cdot \tilde{V}_{F'} = \tilde{V}_{K_1}$ and $\tilde{V}_K \cdot \tilde{V}_{F'} = \tilde{V}_{K_2}$.

Since $\tilde{J}\psi_2 \cdot \beta = \tilde{F}\psi_1 \cdot \alpha$ in $\mathcal{C}^{\mathcal{P}_0}$, and $\tilde{V}_{F'} \cdot \tilde{V}_J = \tilde{V}_{J'} \cdot \tilde{V}_F$, we then have $\tilde{V}_{F'} \tilde{V}_J (\tilde{J}\psi_2 \cdot \beta) = \tilde{V}_{J'} \tilde{V}_F (\tilde{F}\psi_1 \cdot \alpha)$. This equality, together with the naturality of ε^F and ε^J and the way in which the counits of $\tilde{V}_{F'}$ and $\tilde{V}_{J'}$ were chosen gives us

$$\varepsilon_{\tilde{K}D_4}^{F'} \cdot \tilde{V}_{F'} \psi_2 \cdot \tilde{V}_{F'} \bar{\beta} = \varepsilon_{\tilde{K}D_4}^{J'} \cdot \tilde{V}_{J'} \psi_1 \cdot \tilde{V}_{J'} \bar{\alpha}.$$

Since by construction $\bar{\delta}$ and $\bar{\gamma}$ are a pushout of $\tilde{V}_{F'} \bar{\beta}$ and $\tilde{V}_{J'} \bar{\alpha}$ in $\mathcal{C}^{\mathcal{P}_3}$, then there exists a unique morphism $\psi : D_3 \rightarrow \tilde{K}D_4$ such that $\varepsilon_{\tilde{K}D_4}^{F'} \cdot \tilde{V}_{F'} \psi_2 = \psi \cdot \bar{\delta}$ and $\varepsilon_{\tilde{K}D_4}^{J'} \cdot \tilde{V}_{J'} \psi_1 = \psi \cdot \bar{\gamma}$.

We have to check that this natural transformation ψ , together with the functor K , is indeed our desired universal morphism. By applying \tilde{F}' to the equality $\varepsilon_{\tilde{K}D_4}^{F'} \cdot \tilde{V}_{F'} \psi_2 = \psi \cdot \bar{\delta}$, and by the naturality of $\eta^{F'}$, we obtain $\tilde{F}'\psi \cdot \delta = \psi_2$. Similarly, by applying \tilde{J}' to $\varepsilon_{\tilde{K}D_4}^{J'} \cdot \tilde{V}_{J'} \psi_1 = \psi \cdot \bar{\gamma}$, and by naturality of $\eta^{J'}$, we obtain the equality $\tilde{J}'\psi \cdot \gamma = \psi_1$. Thus, we have a natural transformation $\psi : D_3 \rightarrow \tilde{K}D_4$ such that $\tilde{F}'\psi \cdot \delta = \psi_2$ and $\tilde{J}'\psi \cdot \gamma = \psi_1$.

We still have to show that this ψ is unique satisfying these conditions. Let us assume the existence of another morphism $\xi : D_3 \rightarrow \tilde{K}D_4$ with $\tilde{F}'\xi \cdot \delta = \psi_2$ and $\tilde{J}'\xi \cdot \gamma = \psi_1$. Moving $\tilde{J}\bar{\delta} \cdot \beta = \tilde{F}\bar{\gamma} \cdot \alpha$ to $\mathcal{C}^{\mathcal{P}_3}$, since $\varepsilon^{F'} \cdot \tilde{V}_{F'} \varepsilon_{\tilde{F}'}^J = \varepsilon^{J'} \cdot \tilde{V}_{J'} \varepsilon_{\tilde{J}'}^F$, and using the naturality of $\varepsilon^{F'}$ and ε^F , we have that there are two morphisms $\psi, \xi : D_3 \rightarrow \tilde{K}D_4$ with $\varepsilon_{\tilde{K}D_4}^{F'} \cdot \tilde{V}_{F'} \psi_2 = \psi \cdot \bar{\delta}$ and $\varepsilon_{\tilde{K}D_4}^{J'} \cdot \tilde{V}_{J'} \psi_1 = \psi \cdot \bar{\gamma}$, and $\varepsilon_{\tilde{K}D_4}^{F'} \cdot \tilde{V}_{F'} \psi_2 = \xi \cdot \bar{\delta}$ and $\varepsilon_{\tilde{K}D_4}^{J'} \cdot \tilde{V}_{J'} \psi_1 = \xi \cdot \bar{\gamma}$. Since $\bar{\delta}$ and $\bar{\gamma}$ are a pushout of $\tilde{V}_{F'} \bar{\beta}$ and $\tilde{V}_{J'} \bar{\alpha}$, ψ and ξ must be identical. This means that our construction satisfies the universal property, with (K, ψ) the universal morphism as desired. \square

Lemma 7. For any category \mathcal{C} , $Dg(\mathcal{C})$ has coproducts.

Proof. Let $\{D_i : J_i \rightarrow \mathcal{C}\}_{i \in I}$ be an I -indexed family of diagrams for any set I . Let $\coprod_{i \in I} J_i$ be the coproduct of $\{J_i\}_{i \in I}$ in **Cat**, and let us denote the inclusion morphisms by $\iota_i : J_i \rightarrow \coprod_{i \in I} J_i$.

Note that for $I = \emptyset$, we have that $Dg(\mathcal{C})$ has an initial object. This initial object is the unique functor $\emptyset \rightarrow \mathcal{C}$ with \emptyset the empty category with no objects and no arrows.

We claim that the induced functor $D : \coprod_{i \in I} J_i \rightarrow \mathcal{C}$, with the induced family of inclusion morphisms $\{(\iota_i, 1^{D_i}) : D_i \rightarrow D \cdot \iota_i\}_{i \in I}$, is the coproduct of the indexed family of diagrams $\{D_i\}_{i \in I}$ in $Dg(\mathcal{C})$.

$$\begin{array}{ccc} J_i & \xrightarrow{\iota_i} & \coprod_{i \in I} J_i \\ & \searrow D_i \quad \xRightarrow{1^{D_i}} \quad \swarrow D & \\ & \mathcal{C} & \end{array}$$

Given a diagram $E: A \rightarrow \mathcal{C}$ for some small category A , with an indexed family of morphisms $\{(K_i, \sigma^i): D_i \rightarrow E\}_{i \in I}$, we have to show that there is a unique diagram morphism $(K, \sigma): D \rightarrow E$ such that $(K, \sigma) \cdot (\iota_i, 1^{D_i}) = (K_i, \sigma^i)$ for all $i \in I$.

Since $\coprod_{i \in I} J_i$ is the coproduct of the family $\{J_i\}_{i \in I}$, then there is a unique morphism $K: \coprod_{i \in I} J_i \rightarrow A$ making $K \cdot \iota_i = K_i$.

Let the construction of $\coprod_{i \in I} J_i$ be given by objects (i, a) for $a \in J_i$, and arrows $(i, f): (i, a) \rightarrow (i, b)$ for $f: a \rightarrow b$ in J_i . Then, we have $D(i, a) = D_i(a)$, and $E(K(i, a)) = E(K_i(a))$. Similarly, we have $D(i, f) = D_i(f)$, and $E(K(i, f)) = E(K_i(f))$.

Let us define the natural transformation $\sigma: D \rightarrow E \cdot K$ by choosing

$$\sigma_{(i,a)} = \sigma_a^i: D(i, a) \rightarrow E(K(i, a))$$

for each $(i, a) \in \coprod_{i \in I} J_i$. Its naturality then follows easily from the naturality of the natural transformations $\{\sigma^i: D_i \rightarrow E \cdot K_i\}_{i \in I}$. Furthermore, by construction we have $\sigma \cdot 1^{D_i} = \sigma^i$ for all $i \in I$.

Let us now see that it is unique. Assume another natural transformation $\theta: D \rightarrow E \cdot K$ such that $\theta \cdot 1^{D_i} = \sigma^i$ for all $i \in I$. Since we must have $\sigma_a^i = \theta_{(i,a)} \cdot 1_a^{D_i}$ for all $(i, a) \in \coprod_{i \in I} J_i$, θ must be defined as

$$\theta_{(i,a)} = \sigma_a^i: D(i, a) \rightarrow E(K(i, a))$$

for all $(i, a) \in \coprod_{i \in I} J_i$, and therefore $\sigma = \theta$. \square

4. Structured theories

In this section we define the institution $\mathcal{S}(\mathcal{J})$ of structured theories over a given institution \mathcal{J} , and give some results about the cocompleteness of its categories of signatures and theories, the liberality of $\mathcal{S}(\mathcal{J})$, and the addition of freeness constraints to structured theories.

4.1. The institution of structured theories

As discussed in Section 1, a *structured signature* can be formalized as a functor $D: I \rightarrow \mathbf{Sign}_{\mathcal{J}}$ from a small category I to the category $\mathbf{Sign}_{\mathcal{J}}$ of signatures and signature morphisms in a given institution \mathcal{J} . This is of course a quite general notion. One can specialize the concept to the more familiar concept of hierarchy of signatures by requiring that I is a finite poset and that all the arrows in the diagram are inclusions in an appropriate subcategory of inclusion morphisms. Although it remains to be seen which notion is more useful in practice, we give the constructions for the more general case. We build an institution $\mathcal{S}(\mathcal{J})$, whose theories are called *structured \mathcal{J} -theories*, by defining functors $\text{sen}_{\mathcal{S}(\mathcal{J})}$ and $\text{Mod}_{\mathcal{S}(\mathcal{J})}$ associating to each structured signature D in $\mathbf{Sign}_{\mathcal{S}(\mathcal{J})}$ a set of D -sentences and a category of D -models, respectively. Then, we give a satisfaction relation for it and show that the satisfaction condition holds.

Definition 8. Let us denote by $\mathbf{Sign}_{\mathcal{S}(\mathcal{J})}$ the category $Dg(\mathbf{Sign}_{\mathcal{J}})$ of diagrams over the category of signatures in the institution \mathcal{J} . We shall call the objects of $\mathbf{Sign}_{\mathcal{S}(\mathcal{J})}$ *structured (\mathcal{J} -)signatures*, and will denote each structured signature by its corresponding diagram $D: I \rightarrow \mathbf{Sign}_{\mathcal{J}}$. The morphisms in $\mathbf{Sign}_{\mathcal{S}(\mathcal{J})}$ are called *structured signature morphisms*.

Definition 9. The functor $\text{sen}_{\mathcal{S}(\mathcal{J})}: \mathbf{Sign}_{\mathcal{S}(\mathcal{J})} \rightarrow \mathbf{Set}$, associating to each structured signature $D: I \rightarrow \mathbf{Sign}_{\mathcal{J}}$ a set of sentences and to each structured signature morphism $(K, H): D \rightarrow D'$ a corresponding translation at the level of sentences, is defined as follows:

$$\text{sen}_{\mathcal{S}(\mathcal{J})}(D) = \coprod_{i \in I} \text{sen}_{\mathcal{J}}(D(i)),$$

$$\text{sen}_{\mathcal{S}(\mathcal{J})}((K, H)) = \coprod_{i \in I} \text{sen}_{\mathcal{J}}(H_i).$$

We can see each of the sentences of D as a pair (i, φ) , where φ is a sentence in $\text{sen}_{\mathcal{J}}(D(i))$. Note that, given a structured signature morphism $(K, H): D \rightarrow D'$ and a sentence (i, φ) of D , we have

$$\text{sen}_{\mathcal{S}(\mathcal{J})}((K, H))((i, \varphi)) = (K(i), \text{sen}_{\mathcal{J}}(H_i)(\varphi)).$$

Definition 10. Given a structured signature $D: I \rightarrow \mathbf{Sign}_{\mathcal{J}}$, its category of models $\text{Mod}_{\mathcal{S}(\mathcal{J})}(D)$ has as objects families $M = \{M_i\}_{i \in I}$ with M_i in $\text{Mod}_{\mathcal{J}}(D(i))$, such that for each $\phi: i \rightarrow j$ in I , $\text{Mod}_{\mathcal{J}}(D(\phi))(M_j) = M_i$. A morphism between two such models $f: M \rightarrow M'$ is given by a family $\{f_i: M_i \rightarrow M'_i\}_{i \in I}$ with f_i in $\text{Mod}_{\mathcal{J}}(D(i))$ such that for each $\phi: i \rightarrow j$ in I , $\text{Mod}_{\mathcal{J}}(D(\phi))(f_j) = f_i$.

Definition 11. The functor $\text{Mod}_{\mathcal{S}(\mathcal{J})}: \mathbf{Sign}_{\mathcal{S}(\mathcal{J})} \rightarrow \mathbf{Cat}^{\text{op}}$ assigns to each structured signature $D: I \rightarrow \mathbf{Sign}_{\mathcal{J}}$ its category of models $\text{Mod}_{\mathcal{S}(\mathcal{J})}(D)$, and to each structured signature morphism $(K, H): D \rightarrow D'$ the forgetful functor $\text{Mod}_{\mathcal{S}(\mathcal{J})}((K, H)): \text{Mod}_{\mathcal{S}(\mathcal{J})}(D') \rightarrow \text{Mod}_{\mathcal{S}(\mathcal{J})}(D)$, defined as follows:

$$\text{Mod}_{\mathcal{S}(\mathcal{J})}((K, H))(\{M'_j\}_{j \in I'}) = \{\text{Mod}_{\mathcal{J}}(H_i)(M'_{K(i)})\}_{i \in I},$$

$$\text{Mod}_{\mathcal{S}(\mathcal{J})}((K, H))(\{f'_j\}_{j \in I'}) = \{\text{Mod}_{\mathcal{J}}(H_i)(f'_{K(i)})\}_{i \in I}.$$

Definition 12. Given a structured signature $D: I \rightarrow \mathbf{Sign}_{\mathcal{J}}$, a D -model $M = \{M_i\}_{i \in I}$ satisfies a D -sentence (i, φ) if and only if $M_i \models_{D(i)} \varphi$. In this case, we write $M \models_D (i, \varphi)$.

Proposition 13 (Satisfaction condition). *Let $D: I \rightarrow \mathbf{Sign}_{\mathcal{J}}$ and $D': I' \rightarrow \mathbf{Sign}_{\mathcal{J}}$ be structured signatures, and let $(K, H): D \rightarrow D'$ be a structured signature morphism. Given a D -sentence (i, φ) and a D' -model M' , then*

$$\text{Mod}_{\mathcal{S}(\mathcal{J})}((K, H))(M') \models_D (i, \varphi) \iff M' \models_{D'} \text{sen}_{\mathcal{S}(\mathcal{J})}((K, H))((i, \varphi)).$$

Proof. $\text{Mod}_{\mathcal{S}(\mathcal{J})}((K, H))(M')$ is a D -model, namely, the family of models

$$\text{Mod}_{\mathcal{S}(\mathcal{J})}((K, H))(M') = \{\text{Mod}_{\mathcal{J}}(H_i)(M'_{K(i)})\}_{i \in I}.$$

Since φ is a sentence in $D(i)$, we have

$$\text{Mod}_{\mathcal{S}(\mathcal{J})}((K, H))(M') \models_D (i, \varphi) \iff \text{Mod}_{\mathcal{J}}(H_i)(M'_{K(i)}) \models_{D(i)} \varphi.$$

On the other hand, since $\text{sen}_{\mathcal{S}(\mathcal{J})}((K, H))((i, \varphi)) = (K(i), \text{sen}_{\mathcal{J}}(H_i)(\varphi))$, and thus $\text{sen}_{\mathcal{J}}(H_i)(\varphi)$ is a sentence in $D'(K(i))$, we have

$$M' \models_{D'} \text{sen}_{\mathcal{S}(\mathcal{J})}((K, H))((i, \varphi)) \iff M'_{K(i)} \models_{D'(K(i))} \text{sen}_{\mathcal{J}}(H_i)(\varphi).$$

By the satisfaction condition for the institution \mathcal{J} , we also have

$$\text{Mod}_{\mathcal{J}}(H_i)(M'_{K(i)}) \models_{D(i)} \varphi \iff M'_{K(i)} \models_{D'(K(i))} \text{sen}_{\mathcal{J}}(H_i)(\varphi),$$

and therefore,

$$\text{Mod}_{\mathcal{S}(\mathcal{J})}((K, H))(M') \models_D (i, \varphi) \iff M' \models_{D'} \text{sen}_{\mathcal{S}(\mathcal{J})}((K, H))((i, \varphi)). \quad \square$$

Definition 14. Let $\mathcal{S}(\mathcal{J})$ be the institution with:

- **Sign** $_{\mathcal{S}(\mathcal{J})}$ as category of signatures,
- the sentence functor $\text{sen}_{\mathcal{S}(\mathcal{J})} : \mathbf{Sign}_{\mathcal{S}(\mathcal{J})} \rightarrow \mathbf{Set}$, of Definition 9,
- the model functor $\text{Mod}_{\mathcal{S}(\mathcal{J})} : \mathbf{Sign}_{\mathcal{S}(\mathcal{J})} \rightarrow \mathbf{Cat}^{\text{op}}$, of Definition 11, and
- the satisfaction relation given in Definition 12, for which the satisfaction condition holds as shown in Proposition 13.

Note that the notion of structured \mathcal{J} -theory, that is, of a theory presentation in $\mathcal{S}(\mathcal{J})$, captures well the intuitive notion of structured theory found in actual specifications. Indeed, when a subtheory is imported, its axioms typically *are not repeated again*; they are implicitly inherited from the subtheory. This means that axioms are presented *locally*, for a specific local signature $D(i)$, corresponding to our formal notion of a pair (i, φ) . It also means that at each stage in the specification only the *incremental information* of additional axioms has to be made explicit. This correspondence of our notion with the actual specification practice can be made even more intuitive by remarking that, in practice, theories are typically *named* entities like **BOOL**, **NAT**, **LIST**, etc. Therefore, we should think of the index set $|I|$ in a structured signature $D : I \rightarrow \mathbf{Sign}_{\mathcal{J}}$ as the set of *names* for the different theories present in the structure. Then a sentence such as `nil.L=L` in, say, the theory **LIST** is expressed in our framework as the pair $(\mathbf{LIST}, \text{nil.L=L})$, indicating how the axiom is localized to the **LIST** component of the corresponding structured theory.

Since $\mathbf{Sign}_{\mathcal{S}(\mathcal{J})} = \text{Dg}(\mathbf{Sign}_{\mathcal{J}})$, there should be a close and systematic relationship between the category $\mathbf{Th}_{\mathcal{S}(\mathcal{J})}$ of structured \mathcal{J} -theories in the institution $\mathcal{S}(\mathcal{J})$ and the diagram category $\text{Dg}(\mathbf{Th}_{\mathcal{J}})$. We can express this relationship as an adjunction with particularly good properties.

Let $J : \mathbf{Th}_{\mathcal{S}(\mathcal{J})} \rightarrow \mathbf{Dg}(\mathbf{Th}_{\mathcal{J}})$ be the functor defined on objects by the equality

$$J(D, \Gamma) = D_{\Gamma^*},$$

where if $D : I \rightarrow \mathbf{Sign}_{\mathcal{J}}$ is a structured signature, then $D_{\Gamma^*} : I \rightarrow \mathbf{Th}_{\mathcal{J}}$ has $D_{\Gamma^*}(i) = (D(i), \Gamma_i^*)$ and $D_{\Gamma^*}(\phi : i \rightarrow j) = D(\phi)$, where

$$\Gamma_i^* = \{\varphi \in \text{sen}_{\mathcal{J}}(D(i)) \mid \forall M \in \text{Mod}_{\mathcal{S}(\mathcal{J})}(D, \Gamma), M_i \models_{D(i)} \varphi\}.$$

Note that then $D(\phi) : D_{\Gamma^*}(i) \rightarrow D_{\Gamma^*}(j)$ is indeed a theory morphism because

$$\begin{aligned} \varphi \in \Gamma_i^* &\Leftrightarrow \forall M \in \text{Mod}_{\mathcal{S}(\mathcal{J})}(D, \Gamma), M_i \models_{D(i)} \varphi \\ &\Leftrightarrow (\text{by satisfaction, with } \text{Mod}_{\mathcal{J}}(\phi)(M_j) = M_i) \\ &\quad \forall M \in \text{Mod}_{\mathcal{S}(\mathcal{J})}(D, \Gamma), M_j \models_{D(j)} \text{sen}_{\mathcal{J}}(D(\phi))(\varphi) \\ &\Leftrightarrow \text{sen}_{\mathcal{J}}(D(\phi))(\varphi) \in \Gamma_j^* \end{aligned}$$

Therefore, $\text{sen}_{\mathcal{J}}(D(\phi))(\Gamma_i^*) \subseteq \Gamma_j^*$ and $D(\phi)$ is a theory morphism.

The definition of J on morphisms assigns to each theory morphism $(K, H) : (D, \Gamma) \rightarrow (D', \Gamma')$ in $\mathbf{Th}_{\mathcal{S}(\mathcal{J})}$ the diagram morphism $(K, \tilde{H}) : D_{\Gamma^*} \rightarrow D'_{\Gamma'^*}$ with $\tilde{H}_i = H_i$ for each $i \in I$, which is well defined by observing that $\Gamma'^{\bullet} = \coprod_{j \in I'} \Gamma_j'^*$, and using the fact that (K, H) is a theory morphism, so that for each $(i, \varphi) \in \Gamma$ we have $(K(i), \text{sen}_{\mathcal{J}}(H_i)(\varphi)) \in \Gamma'^{\bullet}$.

Let $R : \mathbf{Dg}(\mathbf{Th}_{\mathcal{J}}) \rightarrow \mathbf{Th}_{\mathcal{S}(\mathcal{J})}$ be the functor defined on objects by the equality

$$R(D) = (\text{sign}_{\mathcal{J}} \cdot D, \coprod_{i \in I} \text{ax}(D(i))),$$

where, for (Σ, Γ) a theory, we use the notation $\text{ax}(\Sigma, \Gamma) = \Gamma$.

Note that for any theory (D, Γ) in $\mathbf{Th}_{\mathcal{S}(\mathcal{J})}$ we have a natural isomorphism $RJ(D, \Gamma) \xrightarrow{\varepsilon_{(D, \Gamma)}} (D, \Gamma)$. Indeed, by construction we have $RJ(D, \Gamma) = (D, \Gamma^{\bullet})$. Therefore, both theories are isomorphic, with the identity signature morphism as the isomorphism. We do not need to define R on morphisms, since such a definition follows automatically from the adjunction result below.

Proposition 15. *The functor $J : \mathbf{Th}_{\mathcal{S}(\mathcal{J})} \rightarrow \mathbf{Dg}(\mathbf{Th}_{\mathcal{J}})$ is full and faithful, with R left adjoint to J and ε as the counit.*

Proof. By Theorem 1 of Section IV.3 in [21], if ε is a counit and is an isomorphism, J is full and faithful. So we just have to check the adjunction.

The unit of the adjunction is the $\mathbf{Dg}(\mathbf{Th}_{\mathcal{J}})$ -morphism $\eta_D = (1_I, \eta) : D \rightarrow JRD$ that is, the diagram morphism

$$\begin{array}{ccc} I & \xrightarrow{1_I} & I \\ & \searrow D \quad \eta \quad \swarrow JRD & \\ & \mathbf{Th}_{\mathcal{J}} & \end{array}$$

where, for $D(i) = (\Sigma_i, \Gamma_i)$, $\eta_i : (\Sigma_i, \Gamma_i) \rightarrow (\Sigma_i, (\coprod_{i \in I} \text{ax}(D(i)))_i^*)$ is the identity signature morphism on Σ_i .

Note that, since

$$\forall \varphi \in \Gamma_i, \forall M \in \text{Mod}_{\mathcal{G}(\mathcal{J})}(\text{sign}_{\mathcal{J}} \cdot D, \coprod_{i \in I} \text{ax}(D(i))), M_i \models_{D(i)} \varphi,$$

we have $\varphi \in (\coprod_{i \in I} \text{ax}(D(i)))_i^*$, and therefore η_i is indeed a theory morphism in $\mathbf{Th}_{\mathcal{J}}$.

Let $(K, \beta) : D \rightarrow J(D', \Gamma')$ be a diagram morphism in $Dg(\mathbf{Th}_{\mathcal{J}})$ with $(D', \Gamma') \in |\mathbf{Th}_{\mathcal{G}(\mathcal{J})}|$. We have to show that there is a unique theory morphism $\overline{(K, \beta)} : RD \rightarrow (D', \Gamma')$ such that

$$\begin{array}{ccc} D & \xrightarrow{\eta_D} & JRD \\ & \searrow_{(K, \beta)} & \downarrow J(\overline{(K, \beta)}) \\ & & J(D', \Gamma') \end{array} \quad \begin{array}{ccc} & & RD \\ & & \downarrow \overline{(K, \beta)} \\ & & (D', \Gamma') \end{array}$$

Uniqueness is clear, because if we have (K, β) with, say,

$$\begin{array}{ccc} I & \xrightarrow{K} & I' \\ & \searrow_D \quad \xRightarrow{\beta} \quad \swarrow & \\ & \mathbf{Th}_{\mathcal{J}} & J(D', \Gamma') \end{array}$$

then the signature morphism part of $\overline{(K, \beta)}$, if it exists, must, by construction, coincide with the morphism

$$(K, \text{sign}_{\mathcal{J}} \cdot \beta) : \text{sign}_{\mathcal{J}} \cdot D \rightarrow \text{sign}_{\mathcal{J}} \cdot J(D', \Gamma').$$

So we just need to check that this signature morphism is indeed a theory morphism

$$\overline{(K, \beta)} : RD \rightarrow (D', \Gamma')$$

in $\mathbf{Th}_{\mathcal{G}(\mathcal{J})}$. That is, for each $(i, \varphi) \in \coprod_{i \in I} \text{ax}(D(i))$ we have to show that

$$\forall M' \in \text{Mod}_{\mathcal{G}(\mathcal{J})}(D', \Gamma'), \quad M'_{K(i)} \models_{D'(K(i))} \text{sen}_{\mathcal{G}(\mathcal{J})}(\beta_i)(\varphi),$$

which is equivalent to showing that

$$\forall M' \in \text{Mod}_{\mathcal{G}(\mathcal{J})}(D', \Gamma'), \quad \text{sen}_{\mathcal{G}(\mathcal{J})}(\beta_i)(\varphi) \in \Gamma'^*_{K(i)}.$$

But by the hypothesis that there is a morphism $(K, \beta) : D \rightarrow J(D', \Gamma')$ we know that

$$\forall \varphi \in \text{ax}(D(i)), \quad \text{sen}_{\mathcal{G}(\mathcal{J})}(\beta_i)(\varphi) \in (\Gamma'^*_{K(i)})^\bullet,$$

where, for (Σ, Γ) a theory in $\mathbf{Th}_{\mathcal{J}}$,

$$\Gamma^\bullet = \{\varphi \in \text{sen}_{\mathcal{J}}(\Sigma) \mid \forall M \in \text{Mod}_{\mathcal{J}}(\Sigma, \Gamma), M \models_{\Sigma} \varphi\}.$$

Then, it is enough to show that $(\Gamma'^*_{K(i)})^\bullet = \Gamma'^*_{K(i)}$. Suppose that $\varphi \in (\Gamma'^*_{K(i)})^\bullet$, then,

$$\forall M \in \text{Mod}_{\mathcal{J}}(D'(K(i))), \quad M \models_{D'(K(i))} \Gamma'^*_{K(i)} \Rightarrow M \models_{D'(K(i))} \varphi.$$

Since, by construction, we have

$$\forall M' \in \text{Mod}_{\mathcal{S}(\mathcal{I})}(D', \Gamma'), \quad M'_{K(i)} \models_{D'(K(i))} \Gamma'_i^*,$$

we then have

$$\forall M' \in \text{Mod}_{\mathcal{S}(\mathcal{I})}(D', \Gamma'), \quad M'_{K(i)} \models_{D'(K(i))} \varphi,$$

and therefore $\varphi \in \Gamma_{K(i)}'^*$, as desired.

We thus have that R is left adjoint to J with η as unit. To see that ε is the counit is straightforward. Given the following situation, $\overline{1_{J(D, \Gamma)}}$ is the counit $\varepsilon_{(D, \Gamma)}$ of the adjunction, which is, obviously, an isomorphism. Since for the signature part $\overline{1_{J(D, \Gamma)}}$ is an identity signature morphism, $\overline{1_{J(D, \Gamma)}}$ coincides with $\varepsilon_{(D, \Gamma)}$, as originally defined, and therefore, ε is the counit as desired. \square

$$\begin{array}{ccc} & J(D, \Gamma) & \\ 1_{J(D, \Gamma)} \nearrow & = & \nwarrow J \cdot \overline{1_{J(D, \Gamma)}} \\ J(D, \Gamma) & \xrightarrow{\eta_{J(D, \Gamma)}} & J R J(D, \Gamma) \end{array} \qquad \begin{array}{ccc} & (D, \Gamma) & \\ & \nwarrow \overline{1_{J(D, \Gamma)}} & \\ & R J(D, \Gamma) & \end{array}$$

4.2. Cocompleteness and liberality

Given the institution $\mathcal{S}(\mathcal{I})$, we now present some results on the cocompleteness of its categories of signatures and theories, and on the liberality of $\mathcal{S}(\mathcal{I})$.

Theorem 16. *If $\text{Sign}_{\mathcal{I}}$ is cocomplete then $\text{Sign}_{\mathcal{S}(\mathcal{I})}$ is cocomplete.*

Proof. Since $\text{Sign}_{\mathcal{S}(\mathcal{I})} = \text{Dg}(\text{Sign}_{\mathcal{I}})$, this follows from Theorem 5. \square

By the following well-known result from [19], it follows that, for any institution \mathcal{I} , its category of theories is cocomplete if its category of signatures is cocomplete.

Theorem 17 (Goguen and Burstall [19]). *If \mathcal{I} is an institution such that $\text{Sign}_{\mathcal{I}}$ is cocomplete, then $\text{Th}_{\mathcal{I}}$ is also cocomplete and the forgetful functor $\text{sign}_{\mathcal{I}} : \text{Th}_{\mathcal{I}} \rightarrow \text{Sign}_{\mathcal{I}}$ preserves colimits.*

Corollary 18. *If $\text{Sign}_{\mathcal{I}}$ is cocomplete then $\text{Th}_{\mathcal{S}(\mathcal{I})}$ is cocomplete, and the functor $\text{sign}_{\mathcal{S}(\mathcal{I})} : \text{Th}_{\mathcal{S}(\mathcal{I})} \rightarrow \text{Sign}_{\mathcal{S}(\mathcal{I})}$ preserves colimits.*

The above result by Goguen and Burstall gives a simple criterion to show when all colimits in the category of theories exist. However, it may not be sufficient. We can have institutions with cocomplete categories of theories whose categories of signatures lack some colimits. For example, if we consider signature morphisms in many-sorted

equational logic that can map *operations* to *terms* (as it is allowed for *views* in OBJ [20] and in Maude [6], for example), the category of signatures fails to be cocomplete. The following shows that, independently of the cocompleteness of $\mathbf{Sign}_{\mathcal{S}(\mathcal{I})}$, if $\mathbf{Th}_{\mathcal{I}}$ is cocomplete then $\mathbf{Th}_{\mathcal{S}(\mathcal{I})}$ is also cocomplete.

Theorem 19. *If $\mathbf{Th}_{\mathcal{I}}$ is cocomplete then $\mathbf{Th}_{\mathcal{S}(\mathcal{I})}$ is cocomplete.*

Proof. If $\mathbf{Th}_{\mathcal{I}}$ is cocomplete, then $Dg(\mathbf{Th}_{\mathcal{I}})$ is cocomplete by Theorem 5. Then, since ε is a natural isomorphism, any diagram Ω in $\mathbf{Th}_{\mathcal{S}(\mathcal{I})}$ is isomorphic to the diagram $RJ\Omega$. But, since R is a left adjoint to J , and therefore preserves colimits, and since $Dg(\mathbf{Th}_{\mathcal{I}})$ has colimits by Theorem 5, we have $R(\text{colim } J\Omega) = \text{colim}(RJ\Omega) = \text{colim}(\Omega)$. \square

The following theorem shows that liberality of an institution \mathcal{I} is inherited by $\mathcal{S}(\mathcal{I})$ under natural conditions.

Theorem 20. *If an institution \mathcal{I} is liberal and exact and $\mathbf{Th}_{\mathcal{I}}$ is cocomplete, then $\mathcal{S}(\mathcal{I})$ is liberal.*

Proof. First of all we observe that, for any $(D, \Gamma) \in |\mathbf{Th}_{\mathcal{S}(\mathcal{I})}|$ we have the isomorphism

$$\text{Mod}_{\mathcal{S}(\mathcal{I})}(D, \Gamma) \simeq \lim(\text{Mod}_{\mathcal{I}} \cdot D_{\Gamma^*}). \quad (\dagger)$$

Indeed, since $\Gamma^* = \coprod_{i \in I} \Gamma_i^*$, (D, Γ) and (D, Γ^*) are isomorphic theories. Therefore, for each $M \in |\text{Mod}_{\mathcal{S}(\mathcal{I})}(D)|$ we have $M \models_D \Gamma$ iff $M \models_D \Gamma^*$ iff for each $i \in I$, $M_i \models_{D(i)} \Gamma_i^*$. Therefore, $M \in |\text{Mod}_{\mathcal{S}(\mathcal{I})}(D, \Gamma)|$ iff

- (i) $\forall i \in I$, $M_i \in |\text{Mod}_{\mathcal{I}}(D(i), \Gamma_i^*)|$, and
 - (ii) $\forall \phi : i \rightarrow j$ in I , $\text{Mod}_{\mathcal{I}}(D(\phi))(M_j) = M_i$,
- and similarly, $f : M \rightarrow M'$ is a morphism in $\text{Mod}_{\mathcal{S}(\mathcal{I})}(D, \Gamma)$ iff
- (i) $\forall i \in I$, $f_i : M_i \rightarrow M'_i$ is a morphism in $\text{Mod}_{\mathcal{I}}(D(i), \Gamma_i^*)$, and
 - (ii) $\forall \phi : i \rightarrow j$ in I , $\text{Mod}_{\mathcal{I}}(D(\phi))(f_j) = f_i$.

The above isomorphism (\dagger) then follows either by an explicit limit construction in **Cat**, or, more easily, by observing that **Cat** is monadic [1] over $\mathbf{Graph} = \mathbf{Set}^{\bullet \rightarrow \bullet}$ (therefore the forgetful functor creates limits [21]) which reduces such a limit construction to a construction of limits in **Set**.

But since $\mathbf{Th}_{\mathcal{I}}$ is cocomplete, the colimit of D_{Γ^*} exists, and by exactness of \mathcal{I} we have the isomorphism

$$\lim(\text{Mod}_{\mathcal{I}} \cdot D_{\Gamma^*}) \simeq \text{Mod}_{\mathcal{I}}(\text{colim } D_{\Gamma^*}),$$

which combined with the isomorphism (\dagger) gives us the isomorphism

$$\text{Mod}_{\mathcal{S}(\mathcal{I})}(D, \Gamma) \simeq \text{Mod}_{\mathcal{I}}(\text{colim } D_{\Gamma^*}).$$

Notice also that, by cocompleteness of $\mathbf{Th}_{\mathcal{J}}$, there is a functor

$$\text{colim} : Dg(\mathbf{Th}_{\mathcal{J}}) \rightarrow \mathbf{Th}_{\mathcal{J}}$$

such that for any morphism $(K, H) : (D, \Gamma) \rightarrow (D', \Gamma')$ in $\mathbf{Th}_{\mathcal{J}(\mathcal{J})}$ we have, thanks to the above isomorphism, the following commutative diagram.

$$\begin{array}{ccc} \text{Mod}_{\mathcal{J}(\mathcal{J})}(D, \Gamma) & \xleftarrow{\text{Mod}_{\mathcal{J}(\mathcal{J})}((K, H))} & \text{Mod}_{\mathcal{J}(\mathcal{J})}(D', \Gamma') \\ \wr & & \wr \\ \text{Mod}_{\mathcal{J}}(\text{colim } D_{\Gamma^*}) & \xleftarrow{\text{Mod}_{\mathcal{J}}(\text{colim } J(K, H))} & \text{Mod}_{\mathcal{J}}(\text{colim } D'_{\Gamma'^*}) \end{array}$$

By liberality of the institution \mathcal{J} there is a left adjoint to $\text{Mod}_{\mathcal{J}}(\text{colim } J(K, H))$, which, composed with the vertical isomorphisms, gives rise to a left adjoint to $\text{Mod}_{\mathcal{J}(\mathcal{J})}((K, H))$. \square

4.3. Freeness constraints

One of the key motivations for making structured theories a direct object of study is dealing with *freeness constraints*, called *constraints* or *data constraints* in [19]. They are crucial for the notion of *parameterized module*, in which the model of the parameterized module's body should be a *free extension* of the model of the parameter theory. In many specification languages (e.g., [6,8,9,15,20]) this leads to a distinction between *theories*, with loose semantics, and *modules*, with initial or, more generally, free extension semantics. Both theories and modules can be parameterized, but in the case of parameterized modules, a freeness constraint between models of the parameter and models of the body is enforced.

Intuitively, freeness constraints are associated to particular theory maps appearing in the diagram of a structured theory. Suppose that \mathcal{J} is liberal and that (D, Γ) is a structured theory with $D : I \rightarrow \mathbf{Sign}_{\mathcal{J}}$, and consider a morphism $\phi : i \rightarrow j$ in I . Then, we can associate a freeness constraint to the theory map $D(\phi) : D_{\Gamma^*}(i) \rightarrow D_{\Gamma^*}(j)$ by requiring that the models M of (D, Γ) , in addition to satisfying the axioms Γ , satisfy the constraint

$$M_j \simeq F_{D(\phi)}(M_i)$$

for $F_{D(\phi)} : \text{Mod}_{\mathcal{J}}(D_{\Gamma^*}(i)) \rightarrow \text{Mod}_{\mathcal{J}}(D_{\Gamma^*}(j))$ the left adjoint to the forgetful functor $\text{Mod}_{\mathcal{J}}(D(\phi)) : \text{Mod}_{\mathcal{J}}(D_{\Gamma^*}(j)) \rightarrow \text{Mod}_{\mathcal{J}}(D_{\Gamma^*}(i))$. For example, $D_{\Gamma^*}(i)$ may be the theory TRIV, specifying just one sort `Elt`, and $D_{\Gamma^*}(j)$ may be the theory LIST with a sort `List`, specifying lists formed with data elements from the `Elt` parameter sort. Then, the freeness constraint requires that the models of LIST are really lists, freely generated from the data elements.

The above notion of freeness constraint should in fact be generalized somewhat, to allow an extra signature map bringing the model to the context in which the constraint is applied. This leads us to the following definition, due to Goguen and Burstall.

Definition 21 (Goguen and Burstall [19]). Let \mathcal{J} be an institution. Then a *freeness constraint* on a signature Σ is a pair

$$c = (H : T'' \rightarrow T', G : \text{sign}(T') \rightarrow \Sigma)$$

with H a theory morphism and G a signature morphism. A Σ -model M satisfies c if and only if $\text{Mod}_{\mathcal{J}}(G)(M)$ satisfies T' and $\text{Mod}_{\mathcal{J}}(H)(\text{Mod}_{\mathcal{J}}(G)(M))$ has a free extension along H such that the corresponding component of the counit of the adjunction $\varepsilon_{\text{Mod}_{\mathcal{J}}(G)(M)} : F_H(\text{Mod}_{\mathcal{J}}(H)(\text{Mod}_{\mathcal{J}}(G)(M))) \rightarrow \text{Mod}_{\mathcal{J}}(G)(M)$ is an isomorphism; in this case we write $M \models_{\Sigma} c$.

Our intuitive notion of freeness constraint in a structured theory can then be recovered by two special cases of the above definition. The case of a *parameterized module*, illustrated by the theory inclusion $\text{TRIV} \hookrightarrow \text{LIST}$ in our previous example, corresponds to freeness constraints of the form

$$(D(\phi) : D_{\Gamma^*}(i) \rightarrow D_{\Gamma^*}(j), 1_{D(j)} : D(j) \rightarrow D(j)),$$

whereas the case of an *unparameterized module*, like NAT or BOOL , for which we want an *initial model semantics*, corresponds to a freeness constraint of the form

$$(\emptyset_{D_{\Gamma^*}(j)} : \emptyset \rightarrow D_{\Gamma^*}(j), 1_{D(j)} : D(j) \rightarrow D(j)),$$

where \emptyset is the initial object in the category of signatures of an exact liberal institution \mathcal{J} (so that $\text{Mod}_{\mathcal{J}}(\emptyset)$ has only one model, let us call it also \emptyset) because then the initial model of $D_{\Gamma^*}(j)$ coincides with $F_{\emptyset_{D_{\Gamma^*}(j)}}(\emptyset)$.

The need for the more general notion of freeness constraint in Definition 21 has to do with translation of constraints by composition with signature morphisms. We think of a constraint $c = (H : T'' \rightarrow T', G : \text{sign}(T') \rightarrow \Sigma)$ on Σ as a *sentence* associated to the signature Σ . Then, if $Q : \Sigma \rightarrow \Omega$ is a signature morphism, we can associate to the constraint c the following constraint on Ω :

$$\text{sen}_{\mathcal{J}}(Q)(c) = (H : T'' \rightarrow T', Q \cdot G : \text{sign}(T') \rightarrow \Omega).$$

The key point is that, as shown by Goguen and Burstall [19], the satisfaction condition holds for freeness constraints translated along signature morphisms. Goguen and Burstall exploit this satisfaction condition to give a general construction associating to an institution \mathcal{J} another institution $\mathcal{C}(\mathcal{J})$ (cf. [19, Proposition 23]) with the same category of signatures and the same model functor as \mathcal{J} , and with $\text{sen}_{\mathcal{C}(\mathcal{J})}(\Sigma)$ the disjoint union of the sets $\text{sen}_{\mathcal{J}}(\Sigma)$ and of the set⁴ of all freeness constraints on Σ . Then, by the general result in Theorem 17, if $\mathbf{Sign}_{\mathcal{J}}$ is cocomplete, then $\mathbf{Th}_{\mathcal{C}(\mathcal{J})}$ is a cocomplete category.

Although the construction of $\mathcal{C}(\mathcal{J})$ is given by Goguen and Burstall for an institution \mathcal{J} whose signatures and theories are unstructured, we have pointed out above

⁴ There are foundational questions about the size of the closure of a constraint theory that we will ignore here; as pointed out in [19], they can be solved, for example, by limiting the size of the category of signatures used in the original institution.

how the notion of freeness constraint finds its natural home as additional constraints added to specific components of a structured theory. The way of explicitly combining freeness constraints and structured theories is then straightforward.

Definition 22. Given an institution \mathcal{I} , the institution of *structured \mathcal{I} -theories with freeness constraints* is by definition the institution $\mathcal{S}(\mathcal{C}(\mathcal{I}))$.

Theories in $\mathcal{S}(\mathcal{C}(\mathcal{I}))$ are pairs (D, Γ) , with $D: I \rightarrow \mathbf{Sign}_{\mathcal{I}}$ a diagram, and with Γ sentences of the form (i, φ) , with φ either a sentence in $\text{sen}_{\mathcal{I}}(D(i))$ or a freeness constraint on the signature $D(i)$. Therefore, structured \mathcal{I} -theories with freeness constraints capture the distinction between *theories* (with loose semantics) and *modules* (with initial or free extension semantics) present, as already mentioned, in many algebraic specification languages. Furthermore, they also capture the fact that such theories and modules can be combined into more general *structured specifications with freeness constraints*, whose semantics explicitly depends on their structure.

Notice that, although they are of course related constructions, the institution $\mathcal{S}(\mathcal{C}(\mathcal{I}))$ defined above is different from the institution $\mathcal{C}(\mathcal{S}(\mathcal{I}))$, that, by the general construction of $\mathcal{C}(\mathcal{I})$ of Goguen and Burstall, can also be defined for any institution \mathcal{I} . Intuitively speaking, in $\mathcal{S}(\mathcal{C}(\mathcal{I}))$ the freeness constraints are *local* to specific components of structured theories, whereas in $\mathcal{C}(\mathcal{S}(\mathcal{I}))$ the freeness constraints are *global*, in the sense of involving pairs of structured theories. $\mathcal{S}(\mathcal{C}(\mathcal{I}))$ seems more useful in practice, but the relationship between these institutions and other combinations of the \mathcal{C} and \mathcal{S} constructions should be further studied. Notice also that, by Theorems 16 and 17, if $\mathbf{Sign}_{\mathcal{I}}$ is cocomplete, then both $\mathbf{Th}_{\mathcal{S}(\mathcal{C}(\mathcal{I}))}$ and $\mathbf{Th}_{\mathcal{C}(\mathcal{S}(\mathcal{I}))}$ are also cocomplete.

5. Structured theories in practice

In this section we illustrate the use of the categorical constructions presented in the previous sections by giving several examples of structured theories. We use for that the Maude language [6], and in particular its membership equational logic institution [26]. We present structured theories as diagrams of equational theory *inclusions*, leaving implicit the application of the functor $R: Dg(\mathbf{Th}_{\mathcal{I}}) \rightarrow \mathbf{Th}_{\mathcal{I}(\mathcal{I})}$.

Specifically, we present the equational theory of semimodules over a semiring as a structured theory, which is parameterized by the theory of semirings and the theory of commutative monoids. In the construction of the structure of the theory SEMIMODULE we encounter specifications of theories for monoids, groups, commutative monoids, semirings, and monoid actions.

Let us begin by introducing the functional theory TRIV, which requires just a sort.

```
fth TRIV is
  sort Elt .
endfth
```

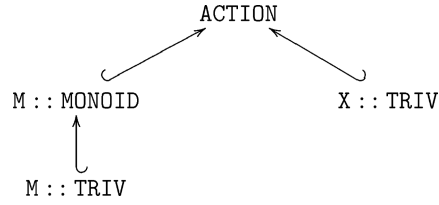


Fig. 1. Structure of the parameterized theory ACTION.

The theory of monoids, with an associative binary operator with identity element 1, can be expressed as follows.

```

fth MONOID is including TRIV .
  op 1 : -> Elt .
  op _ : Elt Elt -> Elt [assoc id: 1] .
endfth
  
```

Next, we define the theory of (left) actions of a monoid on a set. We define it as a functional theory parameterized by the theories MONOID and TRIV.⁵

```

fth ACTION[M :: MONOID, X :: TRIV] is
  op _ : Elt.M Elt.X -> Elt.X .
  vars A B : Elt.M .
  var Y : Elt.X .
  eq 1 Y = Y .
  eq (A B) Y = A (B Y) .
endfth
  
```

Representing by \hookrightarrow the inclusion relations between theories, we can depict the structure of the parameterized theory ACTION as in Fig. 1.

The theory of commutative monoids can be defined just as the theory of monoids, but the $_{+}$ operator is now declared associative, commutative, and has 0 as its identity element.

```

fth +MONOID is including TRIV .
  op 0 : -> Elt .
  op _+_ : Elt Elt -> Elt [assoc comm id: 0] .
endfth
  
```

The theory of semirings can be expressed as follows.

```

fth SEMIRING is
  including MONOID .
  including +MONOID .
  vars X Y Z : Elt .
  
```

⁵ Note the use of the labels associated to the parameters to qualify the sorts coming from the parameter theories.

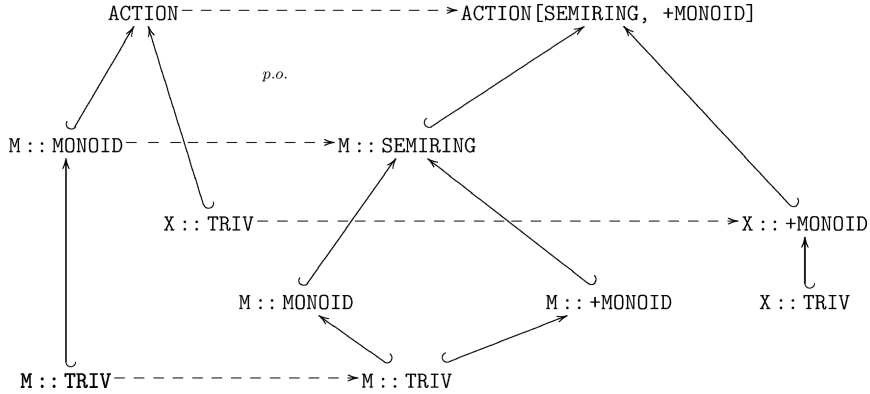


Fig. 2. Instantiation of the parameterized theory ACTION.

```

eq X (Y + Z) = (X Y) + (X Z) .
eq (X + Y) Z = (X Z) + (Y Z) .
eq 1 X = X .
endfth

```

The instantiation of a parameterized theory requires the definition of a *view*, that is, a theory morphism for each of the formal parameters. Given a theory T which is included in another theory T' , let us adopt the convention of naming the view from T to T' defined by the inclusion $T \hookrightarrow T'$ by the name of the “supertheory” T' .

Given the theory ACTION above, the result of instantiating it with views SEMIRING and +MONOID is a theory with name ACTION[SEMIRING, +MONOID] and interface $[M :: SEMIRING, X :: +MONOID]$.

The semantics of the instantiation of theories is given by the pushouts in the category of structured theories discussed in Section 4.2, which can be obtained, using the functor J , from pushouts in $Dg(\mathbf{Th}_{\mathcal{S}})$ thanks to Theorem 19. We can depict the instantiation of the theory ACTION by views SEMIRING and +MONOID by the diagram in Fig. 2. The structured parameterized theory ACTION is understood as the inclusion of its interface $[M :: MONOID, X :: TRIV]$ —corresponding in the figure to the structured theory with tops $M :: MONOID$ and $X :: TRIV$ —into the structured theory with top ACTION. We then perform the pushout of this inclusion along a structured theory map from the interface $[M :: MONOID, X :: TRIV]$ to the structured theory with tops $M :: SEMIRING$ and $X :: +MONOID$ defined by the views SEMIRING and +MONOID.

Theories instantiated by other theories could still be further instantiated. To ‘link’ the formal parameters of a parameterized theory to the formal parameters of another parameterized theory being imported, we use the labels in the interface of the theory, in the OBJ3 style [20], as indicated in the following theory SEMIMODULE. In fact, we can see the labels in the interface as denoting identity views for the theories with which they are associated. For example, given the parameter $S :: SEMIRING$ below, the label S denotes the identity view for the theory SEMIRING, or more precisely, it denotes the view from the structured theory $M :: SEMIRING$ to the structured theory

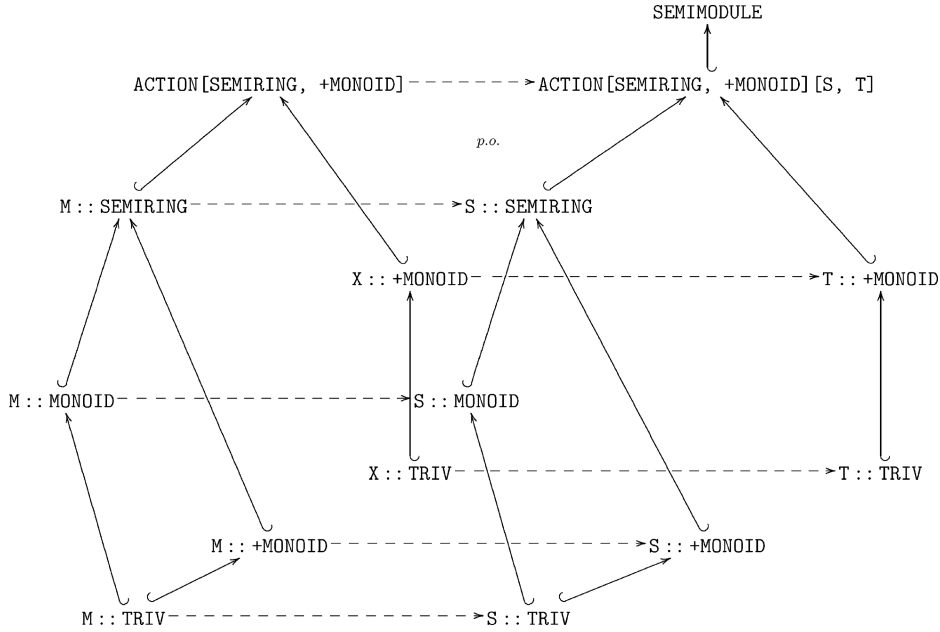


Fig. 3. The structured theory SEMIMODULE.

$S :: \text{SEMIRING}$, which is just an isomorphic copy obtained by renaming.

```

fth SEMIMODULE[S :: SEMIRING, T :: +MONOID] is
  including ACTION[SEMIRING, +MONOID][S, T] .
  vars A B : Elt.S .
  vars X Y : Elt.T .
  eq X (A + B) = (X A) + (X B) .
  eq (X + Y) A = (X A) + (Y A) .
  eq X 0 = 0 .
endfth

```

Linking the parameters of the theory $\text{ACTION}[\text{SEMIRING}, +\text{MONOID}]$ to the formal parameters of the structured theory SEMIMODULE in which it is included, or, in other words, instantiating it with views S and T , results in a structured theory with name $\text{ACTION}[\text{SEMIRING}, +\text{MONOID}][S, T]$ and interface $[S :: \text{SEMIRING}, T :: +\text{MONOID}]$. The structure of the theory SEMIMODULE and the different instantiations involved are depicted in Fig. 3.

6. Concluding remarks

We have shown that the addition of structured theories to an institution \mathcal{I} results in an institution $\mathcal{P}(\mathcal{I})$, and that if the category of signatures $\mathbf{Sign}_{\mathcal{I}}$ has colimits, then the

categories of signatures and theories of $\mathcal{S}(\mathcal{I})$ both have colimits. We have also shown other basic results about the category of theories of $\mathcal{S}(\mathcal{I})$, and about the liberality of the institution $\mathcal{S}(\mathcal{I})$. Finally, we have presented a very simple way of adding freeness constraints to our setting, resulting in institutions $\mathcal{S}(\mathcal{C}(\mathcal{I}))$ and $\mathcal{C}(\mathcal{S}(\mathcal{I}))$.

The notion of structured theory is useful not only for institutions, but also for other components of a logic such as entailment systems or proof calculi [24], and could be naturally extended to those contexts. As already mentioned, the notions presented in this paper can be specialized to the more familiar case of finite hierarchies of theory inclusions by considering diagrams whose diagram schemes are finite posets, and assuming a subcategory of theory inclusions stable under pushouts along the lines of [10]. We think that it is also quite promising to study *heterogeneous* structured theories, involving several institutions, following the heterogenous specification ideas of Tarlecki [33].

As illustrated by the examples in Section 5, the concepts and results presented here have been used as a foundation for the module algebra of the Maude language, in which structured theories are indeed first class citizens [12,13]. Maude is based on rewriting logic [25] and has an equational specification sublanguage based on membership equational logic [26]. In general, theories, structured theories, theory maps, and so on are *metalevel entities* not available at the object level of a logic. However, rewriting logic is reflective [5,7] in the precise sense of having a *universal theory* U such that for any (finitely presented) rewrite theory T and sentence ϕ in T there is a sentence $T \vdash \phi$ in U such that

$$T \vdash \phi \iff U \vdash \overline{T \vdash \phi}.$$

In particular, the rewrite theory U has a sort `Module` whose terms are representations \tilde{T} of rewriting logic theories T . Maude's module algebra is then a rewrite theory `FULL-MAUDE` extending the universal theory U with new sorts such as `StrTheory`, `View`, and so on, and such that all module operations are *definable within the logic by reflection*. That is, in `FULL-MAUDE`, all metalevel entities such as theories, structured theories, views, and module expressions are *reified* at the object level of the logic. Furthermore, the formal specification of the module operations is *executable*, so that `FULL-MAUDE` is part of the Maude distribution and is used in practice to execute the module composition operations.

We plan to study further the institution $\mathcal{S}(\mathcal{C}(\mathcal{I}))$ (and other combinations of the \mathcal{C} and \mathcal{S} constructions) which can serve as a semantic basis for an executable *generic module algebra* that could be specified and executed in Maude, and could be instantiated for one's logic of choice, generalizing Maude's module algebra, which manipulates structured rewrite theories and which, as mentioned above, is expressed and executed within the reflective logical framework of rewriting logic [12,13]. This will allow endowing a specification language of choice with structured theories and with a module algebra for free. Regarding $\mathcal{S}(\mathcal{C}(\mathcal{I}))$, two important questions are: (1) finding appropriate “normal forms” for freeness constraints under suitable assumptions such as persistence; and (2) finding suitable inductive inference systems that, in spite of their intrinsic incompleteness, can approximate the logic of $\mathcal{S}(\mathcal{C}(\mathcal{I}))$ for a given \mathcal{I} .

Acknowledgements

We are grateful to Narciso Martí-Oliet and Peter Mosses for their careful reading of a previous version of this paper and for many useful suggestions to improve the exposition.

References

- [1] M. Barr, C. Wells, *Toposes, Triples and Theories*, Springer, Berlin, 1985.
- [2] H. Baumeister, Unifying initial and loose semantics of parameterized specifications in an arbitrary institution, in: S. Abramsky, T. Maibaum (Eds.), *Proc. Internat. Conf. on Theory and Practice of Software Development (TAPSOFT'91)*, Lecture Notes in Computer Science, Vol. 493, Springer, Berlin, 1991, pp. 103–120.
- [3] M. Bidoit, H.-J. Kreowski, P. Lescanne, F. Orejas, D. Sannella (Eds.), *Algebraic System Specification and Development. A Survey and Annotated Bibliography*, Lecture Notes in Computer Science, Vol. 501, Springer, Berlin, 1991.
- [4] R. Burstall, J. Goguen, The semantics of Clear, a specification language, in: D. Bjørner (Ed.), *Proc. 1979 Copenhagen Winter School on Abstract Software Specification*, Lecture Notes in Computer Science, Vol. 86, Springer, Berlin, 1980, pp. 292–332.
- [5] M. Clavel, *Reflection in general logics and in rewriting logic with applications to the Maude language*, Ph.D. Thesis, Universidad de Navarra, 1998.
- [6] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, J. Quesada, Maude: specification and programming in rewriting logic, *Theoret. Comput. Sci.* 285 (2002) 187–243.
- [7] M. Clavel, J. Meseguer, Reflection and strategies in rewriting logic, in: J. Meseguer (Ed.), *Proc. 1st Internat. Workshop on Rewriting Logic and its Applications*, Electronic Notes in Theoretical Computer Science, Vol. 4, Elsevier, Amsterdam, 1996, available at <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [8] CoFI Task Group on Language Design, CASL—The common algebraic specification language, version 1.0, available at <http://www.brics.dk/Projects/CoFI>, October 1998.
- [9] R. Diaconescu, K. Futatsugi, *CafeOBJ Report*, AMAST Series, World Scientific, Singapore, 1998.
- [10] R. Diaconescu, J. Goguen, P. Stefaneas, Logical support for modularisation, in: G. Huet, G. Plotkin, C. Jones (Eds.), *Proc. of Workshop on Logical Frameworks*, Cambridge University Press, Cambridge, 1991, pp. 83–130.
- [11] T. Dimitrakos, Parameterising (algebraic) specifications on diagrams, in: *Proc. of 13th IEEE Internat. Conf. on Automated Software Engineering (ASE'98)*, 1998.
- [12] F. Durán, *A reflective module algebra with applications to the Maude language*, Ph.D. Thesis, Universidad de Málaga, June 1999.
- [13] F. Durán, J. Meseguer, An extensible module algebra for Maude, in: C. Kirchner, H. Kirchner (Eds.), *Proc. Second Internat. Workshop on Rewriting Logic and its Applications*, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, available at <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [14] F. Durán, J. Meseguer, Structured theories and institutions, in: M. Hofmann, G. Rosolini, D. Pavlović (Eds.), *Proc. 8th Conf. on Category Theory and Computer Science (CTCS'99)*, Electronic Notes in Theoretical Computer Science, Vol. 29, Elsevier, Amsterdam, 1999, pp. 71–90, available at <http://www.elsevier.nl/locate/entcs/volume29.html>.
- [15] H. Ehrig, B. Mahr, *Fundamentals of Algebraic Specification*, Vol. 1, Equations and Initial Semantics, Springer, Berlin, 1985.
- [16] H. Ehrig, B. Mahr, *Fundamentals of Algebraic Specification*, Vol. 2, Module Specifications and Constraints, Springer, Berlin, 1990.
- [17] M. Gogolla, M. Cerioli, What is an abstract data type after all? in: E. Astesiano, G. Reggio, A. Tarlecki (Eds.), *Recent Trends in Data Type Specification*, Lecture Notes in Computer Science, Vol. 906, Springer, Berlin, 1995, pp. 499–523.

- [18] J. Goguen, R. Burstall, Introducing institutions, in: E. Clarke, D. Kozen (Eds.), *Proc. of the Logics of Programming Workshop*, Lecture Notes in Computer Science, Vol. 164, Springer, Berlin, 1984, pp. 221–256.
- [19] J. Goguen, R. Burstall, Institutions: abstract model theory for specification and programming, *J. ACM* 39 (1) (1992) 95–146.
- [20] J. Goguen, T. Winkler, J. Meseguer, K. Futatsugi, J.-P. Jouannaud, Introducing OBJ, Tech. Report SRI-CSL-92-03, Computer Science Laboratory, SRI International, March 1992.
- [21] S. MacLane, *Categories for the Working Mathematician*, Springer, Berlin, 1971.
- [22] N. Martí-Oliet, J. Meseguer, From abstract data types to logical frameworks, in: E. Astesiano, A. Tarlecki (Eds.), *Recent Trends in Data Type Specification*, Lecture Notes in Computer Science, Vol. 906, Springer, Berlin, 1994, pp. 504–514.
- [23] N. Martí-Oliet, J. Meseguer, Rewriting logic as a logical and semantic framework, in: J. Meseguer (Ed.), *Proc. of First Internat. Workshop on Rewriting Logic and its Applications*, *Electronic Notes in Theoretical Computer Science*, Vol. 4, Elsevier, Amsterdam, 1996, available at <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [24] J. Meseguer, General logics, in: H.-D. Ebbinghaus, et al., (Eds.), *Logic Colloquium'87*, North-Holland, Amsterdam, 1989, pp. 275–329.
- [25] J. Meseguer, Conditional rewriting logic as a unified model of concurrency, *Theor. Comput. Sci.* 96 (1992) 73–155.
- [26] J. Meseguer, Membership algebra as a semantic framework for equational specification, in: F. Parisi-Presicce (Ed.), *Recent Trends in Algebraic Development Techniques*, Lecture Notes in Computer Science, Vol. 1376, Springer, Berlin, 1998, pp. 18–61.
- [27] F. Orejas, E. Pino, H. Ehrig, Institutions for logic programming, *Theor. Comput. Sci.* 173 (1997) 485–511.
- [28] H. Reichel, Initially-restricting algebraic theories, in: P. Dembínski (Ed.), *Proc. of Ninth Symp. on Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science, Vol. 88, Springer, Berlin, 1980, pp. 504–514.
- [29] D. Sannella, A. Tarlecki, Specifications in an arbitrary institution, *Inform. and Comput.* 76 (2/3) (1988) 165–210.
- [30] D. Sannella, M. Wirsing, A kernel language for algebraic specification and implementation, in: M. Karpinski (Ed.), *Proc. 1983 Internat. Conf. on Foundations of Computation Theory*, Lecture Notes in Computer Science, Vol. 158, Springer, Berlin, 1983, pp. 415–427.
- [31] H. Schubert, *Categories*, Springer, Berlin, 1972.
- [32] Y. Srinivas, R. Jüllig, SPECWARE: formal support for composing software, in: B. Moeller (Ed.), *Proc. of Conf. on Mathematics of Program Construction*, Lecture Notes in Computer Science, Vol. 947, Springer, Berlin, 1995, pp. 399–422.
- [33] A. Tarlecki, Towards heterogeneous specifications, in: *Proc. of Workshop on Frontiers of Combining Systems (FroCos'98)*, Kluwer Academic, Dordrecht, 1998.